ARMY RESEARCH LABORATORY

Visualization Tool for Guidance Integrated Fuzing

by Geoffrey H. Goldman

ARL-TR-972

March 1996

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1996	Interim, from June to December 95

4. TITLE AND SUBTITLE
Visualization Tool for Guidance Integrated Fuzing

5. FUNDING NUMBERS
DA PR: AH16
PE: P62

6. AUTHOR(S)
Geoffrey H. Goldman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-SE-RM
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-972

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
AMS code: 622120.H1611
ARL PR: 6NE46G

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A software program was developed to visualize a target/missile encounter, overlay point scatterers on target facet models, and display aimpoint tracking positions. It can be used to develop an intuitive understanding of the interaction of various guidance parameters with the target dynamics, which may be most useful for guidance data that resulted in suboptimal fuzing. Guidance input data were generated by MPSIM2 (a mature simulation program for a multi-mode Patriot missile), processed, and then stored. The software was tested on guidance data generated with targets consisting of a single point scatterer.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Guidance integrated fuzing, Patriot, visualization, aimpoint	18
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

Contents

Figures

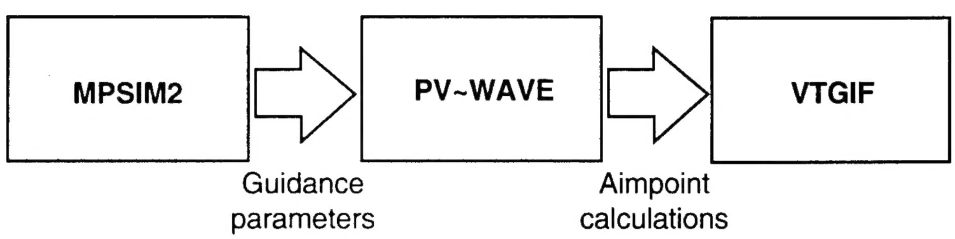
1. Introduction

Missile guidance integrated fuzing (GIF) technology seeks to eliminate the need for a separate on-board fuze by using missile guidance parameters to calculate warhead burst time. Guidance data can be generated from missile seeker simulation code and from missile flight test data. Visualization of a missile/target encounter is an important step in gaining an intuitive understanding of the interaction of guidance parameters that are used to determine warhead burst time. Analysis of missile guidance data that resulted in suboptimal fuzing performance is a first step in improving GIF technology.

2. Software Requirements

This report describes a software program that is a visualization tool for guidance integrated fuzing (VTGIF) data, as well as the accompanying simulation-specific supporting software. The VTGIF software was written in C++, Silicon Graphics (SGI) version 4.0, a graphical user interface (GUI) was implemented with the Open Software Foundation (OSF) Motif toolkit, version 1.2.4, and three-dimensional viewing and picking were implemented with SGI's Open Inventor toolkit, version 2.0.1. VTGIF requires an SGI graphics workstation running an operating system of IRIX 5.3 or higher. Guidance data were generated for VTGIF by the Army's multi-mode Patriot simulation (MPSIM2) program.[1] An intermediate PV~WAVE program (PV~WAVE is a software package used for analyzing scientific data) was required to format and process the guidance data generated by MPSIM2 before it was displayed by VTGIF. Figure 1 shows the processing sequence for the data. Intermediate files were generated to store the data from MPSIM2 and PV~WAVE code.

Figure 1. Processing block diagram for guidance data.



[1]*MPSIM2 Analyst Manual, U.S. Army Missile Command, Redstone Arsenal, Alabama (August 1993).* (CONFIDENTIAL)

3. Simulated Guidance Data

The MPSIM2 program was selected for generating guidance data because it is a mature GIF simulation program that has been tested and validated. The guidance parameters generated by MPSIM2 that were required by VTGIF are missile and target position with respect to the launch location; missile yaw, pitch, and roll; missile antenna azimuth and elevation gimbal angles with respect to the missile body; target azimuth and elevation orientation angles with respect to the missile antenna; simulated azimuth and elevation boresight errors; simulated filtered (guidance and fuzing algorithms) azimuth and elevation boresight errors; and simulated filtered range from the missile to the target. Also, missile flight time and fuzing range error estimates are recorded, but are used only by VTGIF for readout purposes. A complete description of these parameters can be found in the MPSIM2 Analyst Manual.[1,2]

4. Data Processing

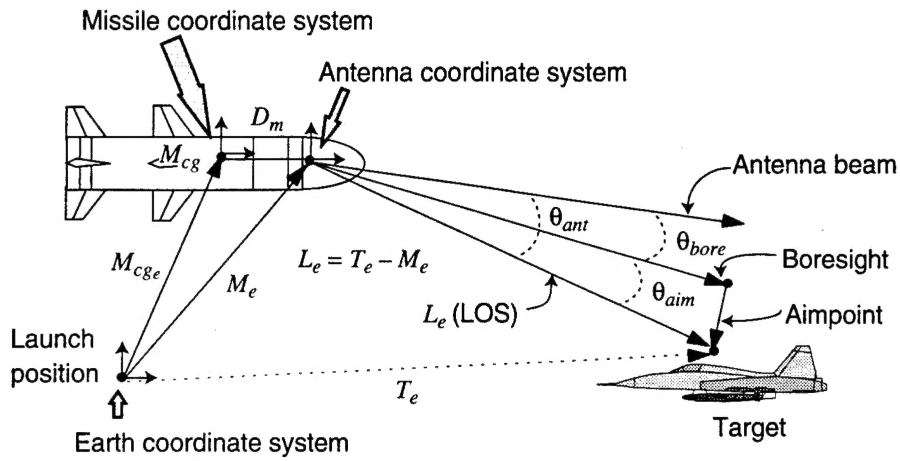
The guidance data and monopulse boresight errors generated by MPSIM2 required processing before aimpoint positions could be graphically displayed on the target. Although this processing could have been incorporated into the VTGIF software, PV~WAVE was used because it simplified the required coding and also provided a hardcopy graphical output. Figure 2 shows a simplified, two-dimensional representation of the typical coordinate systems in a missile/target encounter. If a third dimension were included in the diagram, each angle shown would require an azimuth and an elevation angle component. MPSIM2 uses these general coordinate systems, but with several unique coordinate rotations required to integrate different sections of the software. The processing required to account for these unique rotations is standard, but it is not discussed in this report because it is application specific. The code is provided in the appendix.

The aimpoint of the missile seeker on the target was calculated by two methods. Simulated guidance data could be processed in either the earth (e) or the antenna (a) coordinate system. Several assumptions were made to simplify the calculations. I assumed that the distance from the center of gravity of the missile to the origin of the antenna coordinate system was a constant. This assumption is rea-

[1] *MPSIM2 Analyst Manual, U.S. Army Missile Command, Redstone Arsenal, Alabama (August 1993).* (CONFIDENTIAL)

[2] *Canh Ly, MPSIM Input Namelist Variables, Army Research Laboratory, unpublished (January 1995).*

Figure 2. Coordinate systems of guidance data.



sonable for GIF data, since most or all of the missile's fuel will have been used by the time that fuzing algorithms are engaged. The PV~WAVE code also attempted to synchronize the simulated measurements with the known guidance parameters. An exact synchronization would have required an extensive analysis of the timing diagrams. Therefore, to simplify the analysis, I assumed that there was a fixed processing delay that resulted in a small timing error. I reduced the resulting range error to approximately 1 m by performing an empirically calculated range correction.

The *position* and *vector* notations used below both describe the vectors shown in figure 2. The only difference is that the *vector* notation describes the difference between two guidance parameters, and the *position* notation describes guidance parameters that were relative to the initial launch position of the Patriot missile. In the equations below, matrices start with the letter M and are enclosed by brackets, and variables that can describe both a vector and an angular quantity are denoted with a leading V or θ, respectively. Both processing methods require applying equations (1) to (3) to the required missile and target data to calculate the line of sight (LOS) from the missile to the target.

First, the vector from the missile's center of gravity to the missile's antenna must be transformed to the earth coordinate system:

$$D_e = [M_{ME}]\, D_m \ , \tag{1}$$

where

D_e = vector from the missile center of gravity to the origin of the missile antenna coordinate system, with respect to the earth coordinate system,

M_{ME} = missile to earth rotation matrix, based upon missile yaw, pitch, and roll, and

D_m = vector from the missile center of gravity to the origin of the missile antenna coordinate system, with respect to the missile coordinate system.

To find the position of the missile in the earth coordinate system, we can simply add vectors:

$$M_e = M_{cg} + D_e \ , \tag{2}$$

where

M_e = missile position measured at the origin of the antenna coordinate system, with respect to the earth coordinate system, and

M_{cg} = missile position measured at the center of gravity, with respect to the earth coordinate system.

We can now compute the LOS vector, L_e, as shown in figure 2:

$$L_e = T_e - M_e \ , \tag{3}$$

where

L_e = LOS vector from the antenna to the geometric center of the target, with respect to the earth coordinate system, and

T_e = target position measured at the geometric center of the target, with respect to the earth coordinate system.

For the method based on the antenna coordinate system, equations (4) to (7) are used to calculate the aimpoint vector. First, the boresight error vector is calculated from monopulse angular guidance data:

$$V_{bore_a} = |L| \ [\cos \theta_{az} \cos \theta_{el} , \sin \theta_{az} , \sin \theta_{el}] \ , \tag{4}$$

where

V_{bore_a} = monopulse boresight error vector from the antenna to the aimpoint position, with respect to the antenna coordinate system, labeled "Boresight" in figure 2,

θ_{az} = azimuth angle monopulse boresight error, and

θ_{el} = elevation angle monopulse boresight error.

The boresight error vector is then transformed to the earth coordinate system:

$$V_{bore_e} = [M_{AE}] \ V_{bore_a} \ , \tag{5}$$

where

V_{bore_e} = monopulse boresight error vector from the antenna to the aimpoint position, with respect to the earth coordinate system, and

M_{AE} = antenna to earth rotation matrix, based upon azimuth and elevation missile gimbal angles, and missile yaw, pitch, and roll.

Once transformed to the earth coordinate system, the boresight error vector can be subtracted from the LOS vector:

$$V_{aim_e} = L_e - V_{bore_e} \, , \tag{6}$$

where

V_{aim_e} = aimpoint vector from the target to the boresight vector, with respect to the earth coordinate system, labeled "Aimpoint" in figure 2.

Next, boresight error vector is transformed back to the antenna coordinate system:

$$V_{aim_a} = [M_{EA}] \, V_{aim_e} \, , \tag{7}$$

where

V_{aim_a} = aimpoint vector from the target to the boresight vector, with respect to the antenna coordinate system, and

M_{EA} = earth to antenna rotation matrix, based upon azimuth and elevation missile gimbal angles, and missile yaw, pitch, and roll.

Aimpoint angles can now be determined for the cross range and elevation components of V_{aim_a}:

$$\theta_{aim} = \sin^{-1}(V_{aim_a} / |L|) \, , \tag{8}$$

where

θ_{aim} = azimuth or elevation aimpoint angles, measured with respect to the earth coordinate system.

For the method based on the earth coordinate system, equations (9) to (13) are required to calculate the aimpoint vector. First, the LOS vector is transformed to the antenna coordinate system:

$$L_a = [M_{AE}] \, L_e \tag{9}$$

where

L_a = LOS vector from the antenna to the target, with respect to the antenna coordinate system, and

M_{AE} = antenna to earth rotation matrix based upon azimuth and elevation missile gimbal angles, and missile yaw, pitch, and roll.

The relative antenna beam angles are then determined for the cross range and elevation components of LOS:

$$\theta_{ant} = \sin^{-1}(L_a / |L|) \ , \tag{10}$$

where

θ_{ant} = azimuth or elevation angles between the antenna beam vector and the target vector, with respect to the antenna coordinate system.

Next, the aimpoint angles and vectors are calculated for the cross range and elevation components of V_{aim_a}:

$$\theta_{aim} = \theta_{ant} - \theta_{bore} \ , \tag{11}$$

$$V_{aim_a} = \sin(\theta_{aim}) |L|) \ , \tag{12}$$

$$V_{aim_e} = [M_{EA}] \ V_{aim_a} \tag{13}$$

where

M_{EA} = earth to antenna rotation matrix based upon azimuth and elevation missile gimbal angles, and missile yaw, pitch, and roll.

5. Validation

The aimpoint calculations were validated with known guidance data generated by MPSIM2. A test target model was constructed that consisted of a single point scatterer that changed its position on the target as a function of relative aspect angle with the missile seeker. MPSIM2 was run in a low-noise environment with this target model and several different flight paths. The results indicated that the elevation aimpoint calculations were correct, but the azimuth aimpoint calculations were dependent upon the relative sign of target azimuth aspect angle. The solution was to change the sign of the azimuth aimpoint calculation, which worked satisfactorily. This problem was studied only briefly since the focus of the effort was to visualize GIF data, not to validate MPSIM2 (which is a major task). Figures 3 and 4 show

that the aimpoint position tracks the position of a single scatterer on the test target. Both processing methods produced nearly identical results.

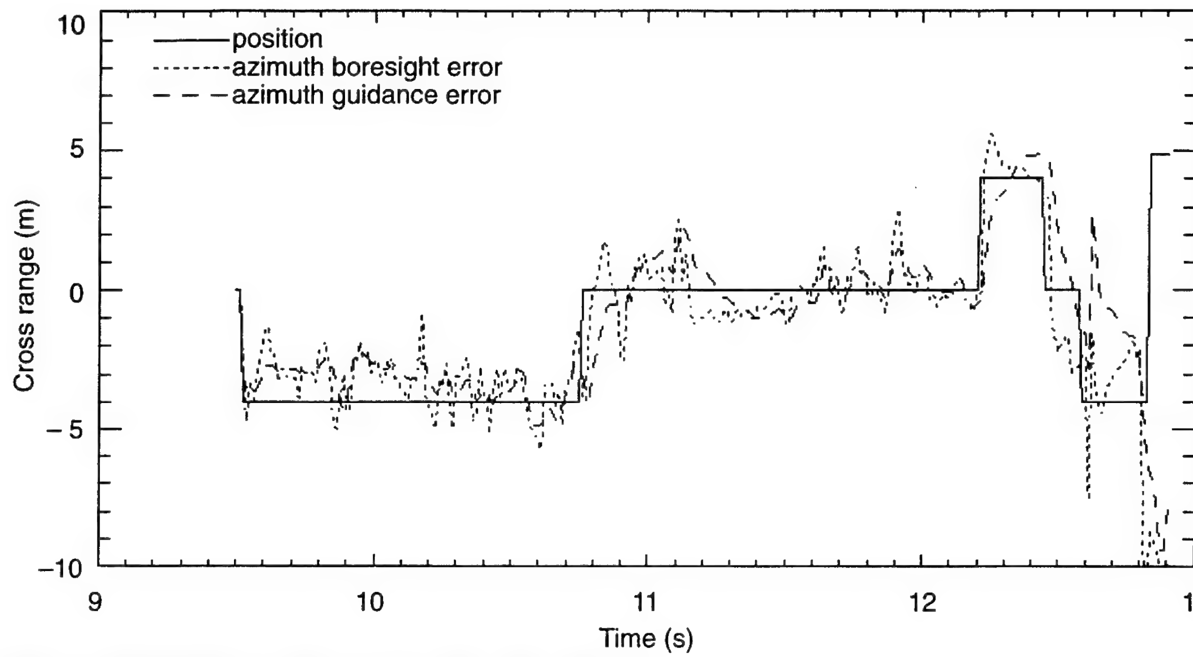


Figure 3. Validation of azimuth aimpoint calculations.

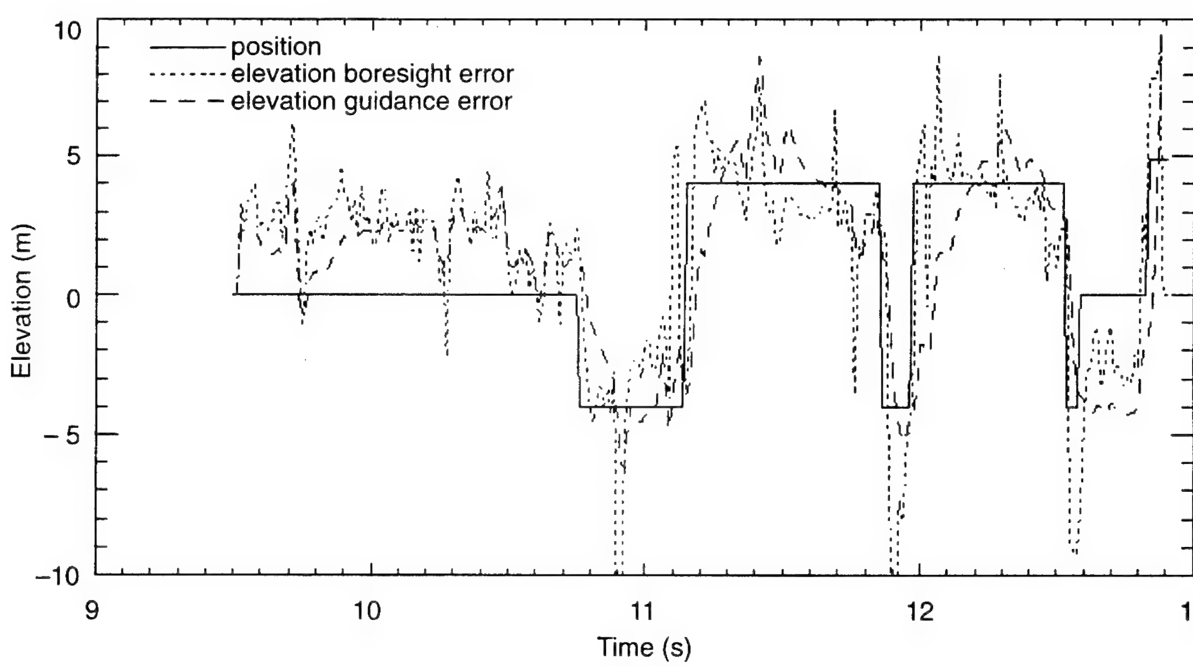


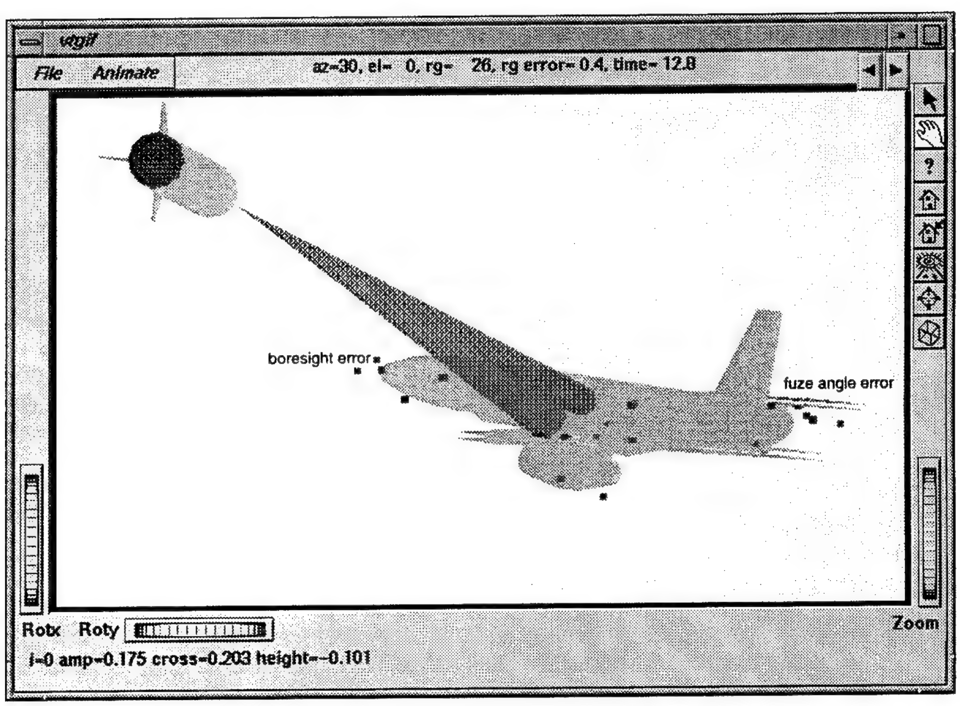
Figure 4. Validation of elevation aimpoint calculations.

6. VTGIF Software

6.1 Introduction

GIF data were visualized with the software program VTGIF, which allows the user to examine the spatial relationships between target aimpoints and scattering centers, and between target and missile positions during a missile/target encounter. The software displays the target, interceptor, antenna beam pattern, aimpoint positions, and target scattering centers. The display window is typically centered on the target, and the target appears to be stationary during most of the encounter. When the target orientation angle changes with respect to the missile seeker by an amount greater than or equal to the angular resolution of the point scatter model, the target will be rotated and the corresponding point scatterers will be displayed. Different target and missile perspectives can be examined via a three-dimensional (3D) viewer provided by the Inventor toolkit. Individual point scatterer radar cross section (RCS) and position can be displayed by the user by "picking" them with the mouse. An extensive collection of viewing formats and options are available to the user to customize the display. Figure 5 shows a snapshot of a single frame from the VTGIF display.

Figure 5. VTGIF display window.



6.2 Menu Selections

The software is controlled through a standard Motif menu interface shown in the top left corner in figure 5. Different geometric target models, point scatterer models, and simulation runs can be selected through the file menu. The geometric target model format was a triangular mesh described by a list of vertex coordinates. The point scatterer model format was target orientation angle data, followed by 50 point scatterers. This format was repeated for each target orientation angle. Each point scatterer consisted of a position in meters and an RCS amplitude in meters. The simulation data consisted of a series of 47 guidance parameters output by the PV~WAVE transformation software. Most of the parameters were not directly used by VTGIF. The encounter could be animated at various speeds selected through the animate menu interface, or stepped through by the user frame by frame. The target orientation, range to the target, range error, and total missile flight time are displayed for each frame.

6.3 3D Viewing

The user can select different 3D viewing perspectives using either the thumb wheels shown in figure 5 or the left mouse button. Different viewing options can be selected from a spring-loaded menu activated by the right mouse button or by the pushbuttons on the right side of the window in figure 5. Extensive on-line help is available for 3D viewing.

7. Conclusion

VTGIF was developed for visualizing a target/missile encounter, overlaying target point scatterers, and displaying aimpoint tracking positions. The software is most useful for targets with distinct regions where scatterers are concentrated. For example, scatterers may be concentrated in the nose section, along specific spots on the body, and on tail fins. Visual analysis can be used to determine which section is being tracked, and what guidance parameters are responsible for a target not being adequately tracked. The code for VTGIF is not included in this report because it is still under development. Future software development may include the use of fragmentation models to animate warhead detonation.

Appendix. Listing of PV~WAVE Code

```
; sanitized PV~WAVE program to process MPSIM GIF data
; antenna coordinate system based-method
; I/O has been eliminated from this code
; potentially sensitive constants have been eliminated from this code

 pro gif

cg_m = [-cg_len, 0 ,0] ; vector from missile cg to antenna

  for i=1, n- delay do begin
    r2 = (data(xm,i- delay) - data(xt,i- delay))^2. + (data(hm,i- delay) -
                data(ht,i- delay))^2. + (data(zm,i- delay) - data(zt,i- delay))^2.

    if r2 LE 0 then rg=1
    rg = r2^.5  ; range from missile to target
    rv = [rg,0,0]

    si = 3.1415926/2.  ; coordinate rotation
    si_90 = [[cos(si), 0, sin(si)], $
            [0,     1, 0],     $
            [-sin(si), 0, cos(si)]]

    fi = pitch_offset
    fi_s = [[1,       0,      0],$
           [0,      cos(fi), -sin(fi)], $
           [0,      sin(fi),  cos(fi)]]

    si = -data(sim,i); yaw
    si_m = [[cos(si), 0, sin(si)], $
            [0,     1, 0],     $
            [-sin(si), 0, cos(si)]]

    theta = -data(thm,i) ; pitch
    theta_m = [[cos(theta), -sin(theta), 0 ], $
                [sin(theta),  cos(theta), 0 ], $
                [0,        0,      1]]
    fi = -data(fim,i) ; roll
    fi_m = [[1,       0,      0],$
             [0,      cos(fi), -sin(fi)], $
             [0,      sin(fi),  cos(fi)]]


    theta = -data(elgs,i) ; antenna elevation gimbal angle
    theta_s = [[cos(theta), -sin(theta), 0 ], $
               [ sin(theta), cos(theta),0 ], $
               [0,       0,     1]]

    si = -data(azgs,i) ; antenna azimuth gimbal angle
    si_s = [[cos(si), 0, sin(si)], $
            [0,     1, 0],     $
            [-sin(si), 0, cos(si)]]


    miss_p_e = [data(xm,i- delay),data(hm,i- delay),data(zm,i- delay)] ; missile position
    targ_e = [data(xt,i- delay),data(ht,i- delay),data(zt,i- delay)]  ; target position

    cg_e = si_90#si_m#theta_m#fi_m#cg_m
    miss_e = cg_e + miss_p_e;      ; corrected missile position
    los_e = targ_e - miss_e       ; line of sight between target and missile

; boresight vectors

    bore_meas_a=[cos(data(az_meas,i))*cos(data(el_meas,i)),sin(data(el_meas,i)),-
```

```
sin(data(az_meas,i))]*rg
    bore_lak_a = [cos(data(az_lak,i))*cos(data(el_lak,i)),sin(data(el_lak,i)),-
sin(data(az_lak,i))]*rg
    bore_fuze_a = [cos(data(az_fuze,i))*cos(data(el_fuze,i)),sin(data(el_fuze,i)),-
sin(data(az_fuze,i))]*rg

;   convert from antenna/seeker frame to earth frame

        bore_meas_e=si_90#si_m#theta_m#fi_m#fi_s#theta_s#si_s#bore_meas_a
        bore_lak_e=si_90#si_m#theta_m#fi_m#fi_s#theta_s#si_s#bore_lak_a
        bore_fuze_e=si_90#si_m#theta_m#fi_m#fi_s#theta_s#si_s#bore_fuze_a

    aim_diff_meas_e = los_e - bore_meas_e
    aim_diff_fuze_e = los_e - bore_fuze_e
    aim_diff_lak_e = los_e - bore_lak_e

    sign=1.
    if data(psif,i) GT 0 then sign = -1.

    aim_diff_meas_e(0) = aim_diff_meas_e(0)*sign
    aim_diff_fuze_e(0) = aim_diff_fuze_e(0)*sign
    aim_diff_lak_e(0) = aim_diff_lak_e(0)*sign

; convert aim_diff_e to aim_diff_a
   si = -3.1415926/2.  ; coordinate rotation
    si_90 = [[cos(si), 0, sin(si)], $
          [0,     1, 0],    $
          [-sin(si), 0, cos(si)]]

    si = data(sim,i); yaw
    si_m = [[cos(si), 0, sin(si)], $
           [0,     1, 0],    $
           [-sin(si), 0, cos(si)]]

    theta = data(thm,i) ; pitch
    theta_m = [[cos(theta), -sin(theta), 0 ], $
               [sin(theta),  cos(theta), 0 ], $
               [0,        0,       1]]

    fi = data(fim,i) ; roll
    fi_m = [[1,       0,      0], $
           [0,      cos(fi), -sin(fi)], $
           [0,      sin(fi),  cos(fi)]]

    fi = -pitch_offset
    fi_s=[[1,       0,      0],$
          [0,      cos(fi), -sin(fi)], $
          [0,      sin(fi),  cos(fi)]]

    theta = data(elgs,i) ; antenna elevation gimbal angle
    theta_s = [[cos(theta), -sin(theta), 0 ], $
               [ sin(theta), cos(theta),0 ], $
               [0,       0,       1]]

    si = data(azgs,i)   ; antenna azimuth gimbal angle
    si_s = [[cos(si), 0, sin(si)], $
           [0,    1, 0], $
           [-sin(si), 0, cos(si)]]

;   convert from earth frame to antenna/seeker frame

        aim_diff_meas_a=si_s#theta_s#fi_s#fi_m#theta_m#si_m#si_90#aim_diff_meas_e;
aim_diff_lak_a=si_s#theta_s#fi_s#fi_m#theta_m#si_m#si_90#aim_diff_lak_e;
aim_diff_fuze_a=si_s#theta_s#fi_s#fi_m#theta_m#si_m#si_90#aim_diff_fuze_e;

; calculate aimpoint angles, and assign values to output array
```

```
    data(az_meas,i) = asin(aim_diff_meas_a(2)/rg)
    data(el_meas,i) = asin(aim_diff_meas_a(1)/rg)

    data(az_fuze,i) = asin(aim_diff_fuze_a(2)/rg)
    data(el_fuze,i) = asin(aim_diff_fuze_a(1)/rg)

    data(az_lak,i) = asin(aim_diff_lak_a(2)/rg)
    data(el_lak,i) = asin(aim_diff_lak_a(1)/rg)

return
end
```

Distribution

Admnstr
Defns Techl Info Ctr
Attn DTIC-OCP
8725 John J Kingman Rd Ste 0944
FT Belvoir VA 22060-6218

US Army MICOM
Attn AMSMI-RD-ST-WF G Henderson
Redstone Arsenal AL 35898-5247

US Army Mis Cmnd
Attn AMSMI-RD-SS J Cole
Attn AMSMI-RD-SS-SI J Jordan
Redstone Arsenal AL 35898

Nav Air Warfare Ctr
Attn Code 472E00D L Stephens
1 Administration Circle
China Lake CA 93555

US Army Rsrch Lab
Attn AMSRL-OP-SD-TA Mail & Records Mgmt
Attn AMSRL-OP-SD-TI Tech Lib (3 copies)
Attn AMSRL-OP-SD-TP Tech Pub (2 copies)
Attn AMSRL-SE J M Miller
Attn AMSRL-SE-R A Sindoris
Attn AMSRL-SE-R E Burke
Attn AMSRL-SE-RM B Wallace
Attn AMSRL-SE-RM C Ly
Attn AMSRL-SE-RM D W Vance
Attn AMSRL-SE-RM G Goldman (20 copies)
Attn AMSRL-SE-RM J Nemarich
Attn AMSRL-SE-RM J Silverstein
Attn AMSRL-SE-RM J Silvious
Attn AMSRL-SE-RM P Cremona-Simmons
Adelphi MD 20783-1197

DEPARTMENT OF THE ARMY
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197